

doi:10.19920/j.cnki.jmsc.2025.02.007

# 面向联合补货 – 配送问题的 $\varepsilon$ -最优算法和对偶下界<sup>①</sup>

王思睿, 王 林\*, 吴彬溶, 张金隆

(华中科技大学管理学院, 武汉 430074)

**摘要:** 针对多物品库存系统中常见的联合补货 – 配送问题, 挖掘了其数学模型中的一系列性质, 并提出了一种  $\varepsilon$ -最优算法和一个对偶下界. 实验结果表明, 本文设计的算法效果超过了现有研究中表现最好的几种元启发式算法, 对目标值的提升幅度最高能达到 31%, 对偶下界与最优值的偏差较小, 平均不超过 0.84%. 新算法运算效率较高, 100 个商品的算例平均运算时间不超过 23 秒. 此外, 本文还对算法时间复杂度进行了分析, 在实际应用环境中, 该算法能够达到  $O(n)$  级多项式时间复杂度, 这一结论可为联合补货 – 配送问题的算法设计理论提供有益的补充. 最后, 对协同配送策略的效益进行了分析. 实验发现, 只有当中心仓库的库存管理水平相比零售商处的库存管理水平有较大优势时, 协同配送才能够取得较高的成本节约效益.

**关键词:** 联合补货 – 配送; Lipschitz 优化;  $\varepsilon$ -最优; 定界启发式算法; 对偶下界

**中图分类号:** F274; F224      **文献标识码:** A      **文章编号:** 1007-9807(2025)02-0085-17

## 0 引 言

联合补货问题 (joint replenishment problem, JRP) 考虑的是从供应商处对多种产品进行分组补货, 从而达到分摊主要准备费用、节省补货总费用的目的. 当一组商品都是由同一供应商或供应地供应, 或当一组商品同时采用一种运输工具运输时, 联合补货有助于降低成本. 在联合补货系统中, 库存管理采取周期盘点策略, 每次订货都会发生主要订货费用与次要订货费用. 主要订货费用只与订货行为相关, 次要订货成本则与商品属性相关 (大小、体积等商品特性), 一次订货的总次要订货成本则是该次订货的所有商品次要订货成本之和. 由于这种特殊的成本结构, 通过合理的安排每次订货时商品的组合能够实现可观的规模效益<sup>[1]</sup>. JRP 自提出以来其学术价值被广泛认同, 具体可以参考 Khouja 和 Goyal<sup>[2]</sup> 的综述性文章, 其应用价值也备受认可, 如沃尔玛和家乐福等国外

知名企业在中国利用联合补货策略来获得物美价廉的商品.

尽管 JRP 模型具有一定的应用价值, 其基本假设较苛刻, 如假设每次补货时发生一次配送行为, 这与连锁零售行业中常见的中心化供应链结构有较大差异. 为更贴近企业管理实践来提升模型的应用价值, 有学者提出通过中心仓库将配送决策集成到 JRP 模型当中, 来协调上下游的补货和配送行为, 这就是联合补货 – 配送 (joint replenishment and delivery, JRD) 问题<sup>[3]</sup>. 这类模型有很强的现实价值: 1) 在制造业和零售业行业中, JRD 模型的三层供应链结构很受青睐. 此类企业往往采取如下物流模式, 即首先通过集中采购的方式从供应商处大批量采购商品, 然后储存在一个中心仓库处等待配送, 并最终配送到终端零售商或客户处. 这种三级供应链结构中的“集中采购策略”便是一种朴素的联合补货思想; 2) 在电力行业, JRD 模型也有很高的应用价值. 比如长江电力

① 收稿日期: 2021-07-30; 修订日期: 2022-12-30.

基金项目: 国家社科基金资助重大项目 (20&ZD126); 中央高校基本业务费资助项目 (2021WKfZZX014).

通讯作者: 王 林 (1974—), 男, 湖北枣阳人, 博士, 教授, 博士生导师. Email: wanglin@hust.edu.cn

公司采购中心面向三峡电厂、葛洲坝电厂、检修厂、梯级调度中心、溪洛渡电厂、向家坝电厂等服务对象,此时下游服务对象可能分布在几百公里的范围内,且物资交付及时性要求高,运输费用不菲.在这种情况下,既要兼顾上游的订货安排,又要注意协同下游的配送调度,给物流与供应链管理带来更大的挑战.因此,对 JRD 研究而言,如何确定中心仓库的最佳服务方案和最优订货策略,是决策者迫切需要解决的问题.

JRD 模型的实用性在过去的研究中也被众多学者广泛证实并得到广泛应用<sup>[4-8]</sup>,但目前对于协同配送策略的效益尚未有深入的分析.对比无协同的 JRP 模型,JRD 的协同配送策略到底效益如何?在何种情况下,协同配送策略才能够取得优势?针对此问题,本文发现了协同配送策略的一个必要边界:只有当中心仓库的库存管理水平高于零售商处的库存管理水平时,协同配送策略才有意义;并且中心仓库的库存管理水平越高,这种协同效益也更突出.

另外,JRD 研究的一个瓶颈就是高质量求解算法的设计,目前不存在有效、通用的精确算法.作为一种强 NP-hard 问题<sup>[9]</sup>,JRD 问题属于混合整数非线性规划,问题难度较大,即使是寻找问题的一个较好的下界也比较困难,目前仅有文献[10]提供了一种获取目标值下界的算法.传统的数学规划方法并不适用,目前比较常见的求解算法是以遗传算法(genetic algorithm, GA)和差分进化算法(differential evolution, DE)为代表的各种改进的元启发式算法.尽管此类算法易于操作,并且都在特定的应用背景下取得了较好的应用效果,但当问题规模扩大或复杂度变高后都不可避免地出现了搜索效率低、算法精度差等问题.

为克服以上困难,本文提出了一种能有效求解 JRD 问题的高质量求解算法:基于 Lipschitz 连续性的  $\varepsilon$ -最优算法.作为一种最优算法,该算法能够保证所得到的目标值与理论最优值的偏差不超过一个任意预设的容差  $\varepsilon$ .在 500 个随机算例的仿真实验中,算法的表现超过了现有研究中常用的、性能较佳的 GA 和 DE 算法,目标值的提升幅度最高能达到 31%,并且算法还有着非常好的运行效率,即使是 100 个商品的大规模算例平均运算时间也不超过 23 s.除此之外,本文还挖掘出

了一系列 JRD 模型的数学性质,包括最优决策变量的必要条件等,并设计了一种定界启发式算法用于缩小问题的搜索空间,对于其他 JRD 扩展问题也有着很好的借鉴意义.另外,本文还通过构造原问题的一个对偶问题,获得了一个对偶下界,仿真算例实验表明该下界与最优值的平均偏差不超过 0.84%,与文献[10]的 21.16%相比有了很大提升,并且与文献[10]复杂的下界算法相比,本文的方法较简单,易于经过少量修改用于其他带资源约束的 JRD 模型求解.

综上所述,本文主要贡献如下:1)证明了 JRD 模型目标函数的 Lipschitz 连续性,提出了一种基于 Lipschitz 优化技术的求解框架;2)挖掘出了基本 JRD 模型的一系列数学性质,并据此设计了一种定界启发式算法,能够有效地寻找决策变量的界限;3)通过构造原问题的对偶问题,给出了原问题的一个较好的下界;4)分析了协同配送策略的效益,发现只有当中心仓库的库存管理水平比零售商有优势时,协同配送策略才有意义.

## 1 文献综述

### 1.1 现有模型的特点与求解方法

作为一个经典的组合优化问题,JRP 一直备受关注.然而经典 JRP 假设每次补货时进行一次配送行为,忽略了配送阶段的优化,故应用价值有限,相关研究可以参考 Khouja 和 Goyal<sup>[2]</sup>的综述性文章.为解决此问题,Cha 等<sup>[3]</sup>提出将 JRP 与单仓库-多零售商问题(one-warehouse N-retailer, OWNR)结合,把配送阶段与补货阶段进行协同优化.具体而言,就是在 JRP 模型中增加中心仓库来充当上下游的“缓冲器”,通过中心仓库来协调补货和配送决策.有无中心仓库的协调配送策略是 JRP 与 JRD 最大的不同之处.由于 JRD 模型更实用,许多学者将其扩展到不同场景下,相继提出更实用的拓展模型.现有研究的关注点主要在于对基础 JRD 模型的扩展,而其求解算法的研究是相对滞后的,主要是各种启发式算法.求解算法的落后也制约了贴近企业管理实践的 JRD 拓展模型的研究,因为这类模型更复杂,求解面临的挑战也更大.在此,本文主要对近几年的 JRD 研究进

行梳理,相关研究模型特点和方法论如表 1 所示.

表 1 近年来 JRD 相关研究汇总分析  
Table 1 JRD researches in recent years

文献	模型特点	方法论
[11]	多供应商选择, 数量折扣	遗传算法
[12]	资源约束, 数量折扣	构造性的启发式算法 混合禁忌搜索算法
[13]	资源约束, 随机需求	遗传算法 Shapley 值法
[14]	资源约束, 自由的配送策略	基于数学规划的元 启发式算法 (Matheuristic)
[15]	越库配送	构造性的启发式算法
[16]	整合运输	量子进化算法
[4]	随机需求, 旅行商策略	自适应混合差分进化算法
[17]	非立即变质品	构造性的启发式算法
[18]	异质品运输	自适应混合差分进化算法
[19]	异质品运输, 商业信用	回溯搜索算法
[20]	有缺陷的商品	遗传算法
[21]	有缺陷的商品, 商品需求替代	差分进化算法
[22]	RFID 投资的影响	差分进化算法
[23]	多中心仓库	自适应混合差分进化算法 混合禁忌搜索算法
[24]	选址决策	遗传算法
[10]	基本 JRD 模型	变邻域搜索算法 界限分析

从模型特点来看,JRD 模型被扩展到了各种各样的应用场景中,大致以下几类:

1) 考虑数量折扣的 JRD. 在公共采购领域,对联合补货和配送集成优化的实例很多,例如目前医药行业兴起的跨区域联合带量采购问题,在这类运营模式中,由一个统一的采购联盟负责管理所有消费者所需物资的采购,在获得数量折扣的情况下通过联合补货策略节省补货成本,文献 [11] 主要聚焦于不同的供应商提供不同的价格折扣时的选择问题,而文献 [12] 则考虑了联合补货阶段和配送阶段的资源约束对数量折扣的影响,他们发现联合补货阶段的资源约束对数量折扣的影响较大.

2) 考虑多种资源约束的 JRD. 除了文献 [12] 之外,还有一些研究关注资源约束,这类应用研究

的应用场景往往见于自建物流的零售企业,此时运输能力的限制对配送和补货的安排有很大影响;文献 [13] 在资源约束的基础上还考虑了随机需求,并且他们还考虑了联合补货联盟的利益分配的博弈问题;文献 [14] 的资源约束与 [12] 类似,但考虑了更为自由的配送方式.

3) 考虑配送方式改进的 JRD. 如文献 [15, 16, 4] 基础的 JRD 模型采取周期式的“一对一”配送方式,文献 [15] 发现,当补货周期和配送周期刚好一致时,可以对终端零售商进行越库配送,从而省下一定的配送费用;文献 [16] 则认为“一对一”的配送方式可以进一步优化,将多个零售商的配送任务打包在一起进行整合运输从而实现规模效益;文献 [4] 则在随机需求的基础上利用旅行商策略来优化配送路径.

4) 考虑特定的商品属性的 JRD. 文献 [17] 考虑了非立即变质品的优化问题,特殊情况下商品可能变质并对库存产生影响;文献 [18] 则考虑了异质品的 JRD 模型,在企业执行 JR 策略时,由于商品种类繁多,往往会出现商品性质相互冲突的情况,例如化工产品和生鲜产品往往不能一起运输,他们的研究考虑了这一影响带来的惩罚;在文献 [18] 的基础上,文献 [19] 额外考虑了企业在应用 JRD 模型时的销售账期问题,他们认为如果不考虑这种账期带来的利息成本和收益,JRD 模型可能会带来额外的利息成本;文献 [20] 则考虑了有缺陷的商品的问题,这是由于多商品运输容易对商品产生损坏进而影响品质;在此基础上,文献 [21] 还考虑到不同商品之间可以相互替代的情况.

5) 考虑多中心仓库和选址问题的 JRD. 文献 [23] 则考虑了存在多个中心仓库的 JRD 问题,通常 JRD 模型假设只有一个中心仓库来运营,而文献 [23] 基于武汉的三个大型仓库的应用情景将问题扩展为了多中心的模型,他们发现多中心模型相较单中心模型存在一定的成本优势;文献 [24] 则进一步地考虑了选址的问题,在他们的模型中可以将客户分配各不同的区域,每一个区域共享一个中心仓库,并采取同样的补货和配送策略.

6) 其它相关主题. 文献 [22] 考虑了企业实行 RFID 技术投资来改进物流效率对 JRD 模型的影

响,正如前文所述,在自建物流的企业中,资源约束的影响是非常大的,而 RFID 投资往往能提高物流效率,变相增加企业的运输资源,他们则考虑了这种投资与带来的产出之间的权衡问题;另外,也有文献[10]关注于基础 JRD 求解算法的设计,其中文献[10]对于 JRD 模型的数学性质有了较多的挖掘,特别是首次证明了 JRD 问题目标函数的一个下界。

尽管 JRD 模型的扩展研究丰富,然而从方法论的角度来看,求解 JRD 的高质量算法研究进展缓慢. Cha 等<sup>[3]</sup>首先提出了一种 GA 算法来求解 JRD,算例实验表明该算法在小规模算例中取得了非常好的效果. 在 Cha 等<sup>[3]</sup>的遗传算法之后,几乎所有的元启发式算法都采取了与其类似的编码方式,只是搜索策略上略有区别,这其中,以差分进化算法及其改进算法取得的效果最为显著<sup>[4, 8, 21-23]</sup>,超过了已有的其他算法,特别是在小规模算例上几乎每次都能够找到最优解,然而,由于搜索空间指数增加,这些算法在大规模问题上的表现都不能够得到保证. 除了元启发式算法之外,文献[12, 15, 17]则采取了构造性的启发式算法,利用到了模型的一些数学结构,尽管运算效率比较高,但仍然不能够保证得到最优解. 就本文整理的文献而言,目前尚未发现 JRD 问题的最优算法的研究.

除了以上整理的优化领域的文献之外,JR 研究的另一个热门领域是联合补货联盟的利益分配博弈问题,联合补货策略的一个现实应用背景是多个企业成立联盟进行采购行为从而节省成本,这种联盟的利益分配也是一个非常有趣的研究方向,这方面的研究见文献[25-29],但这方面的研究主要还是围绕 JRP,未见有 JRD 问题的博弈研究.

### 1.2 现有研究的不足与本文贡献

1) 缺乏最优算法. 目前 JRD 相关的研究更多的是关注于各种类型的扩展应用,主要的求解算法是各类启发式和元启发式算法,目前性能最好的算法是各种改进的差分进化算法,没有最优算法的研究. 本研究则首次提出了一种  $\varepsilon$ - 最优算法,相较启发式算法,该算法求解精度高.

2) 基础 JRD 模型的数学性质挖掘不深. 由于大多数研究关注于 JRD 模型的扩展,却没有讨论 JRD 模型的数学性质,相应的算法设计更是因此缺少理论支撑,本研究不仅证明目标函数的 Lips-

chitz 连续性,更给出了许多有用的数学性质以供参考.

3) 缺少较好的下界算法. 由于 JRD 问题属于强 NP-hard 问题,求解难度较大,即使是目标函数的下界也难以获取,只有文献[10]提出了一种下界算法,而其下界算法较为复杂难以扩展到一般情况,并且在某些情况下的性能表现一般,这就给算法性能评价带来了一定的困难. 在以往的研究中只有小规模算例可以通过枚举法获取最优解,而大规模算例则只能选择现有最好算法的结果作为评价标杆. 本文的实验表明在大规模算例上现有算法的表现与最优解的偏差较大,而本文提出的下界算法实现简单,且与最优解偏差很小,可以有效地评估 JRD 问题求解算法的性能.

## 2 问题描述和数学模型

数学符号如表 2 所示.

表 2 符号标示

Table 2 Notations used in the model

变量/索引	含义描述
$D_i$	商品 $i$ 的需求率
$S$	主要订货成本
$s_i^w$	商品 $i$ 的次要订货成本
$h_i^w$	商品 $i$ 在中心仓库的单位时间库存成本
$s_i^r$	商品 $i$ 的配送成本
$h_i^r$	商品 $i$ 在零售商处的单位时间库存成本
$i$	商品种类索引, $i \in N$
$n$	商品种类数
$N$	商品种类集, 等于 $\{1, 2, \dots, n\}$
决策变量	含义描述
$T$	基本补货周期
$k_i$	控制中心仓库对商品 $i$ 补货的频率
$K$	包括所有 $k_i$ 的向量
$f_i$	控制中心仓库对商品 $i$ 配送的频率
$F$	包含所有 $f_i$ 的向量

本研究采用的多层供应链结构如图 1 所示,由 1 个中心仓库、 $n$  个供应商、 $n$  个零售商构成. 一个常见的应用场景如下:所有的供应商处于同一个地区 A,而中心仓库与零售商则位于地区 B,此时所有零售商的商品需求可以通过中心仓库进行整合,通过同一个交通工具进行运输,从而实现联合补货. 在此模式下,中心仓库需要根据下游零

售商需求信息来协调补货和配送方案,从而实现包括补货、库存和配送费用在内的系统总成本最低.该模型假设供应商和零售商的数量相对应,并且一个零售商只从一个供应商处进货.根据 Wang 等<sup>[10]</sup>的证明,该模型是一个通用模型,其他复杂情况都可以转化为上述供应商与零售商一对一的情况.

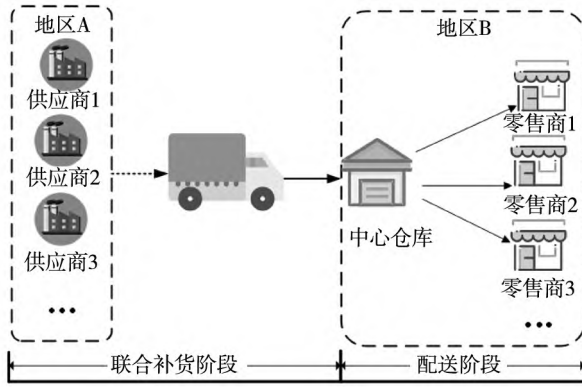


图1 JRD 模型示意图

Fig. 1 The structure of JRD

JRD 问题的优化目标为最小化整个系统单位时间内的平均成本,总成本包括4个部分:订货成本  $C_o$ , 中心仓库库存成本  $C_h^w$ , 配送成本  $C_d$ , 零售商的库存成本  $C_h^r$ . JRD 运作模式下,每个商品在中心仓库和零售商的库存量变化趋势见图2.

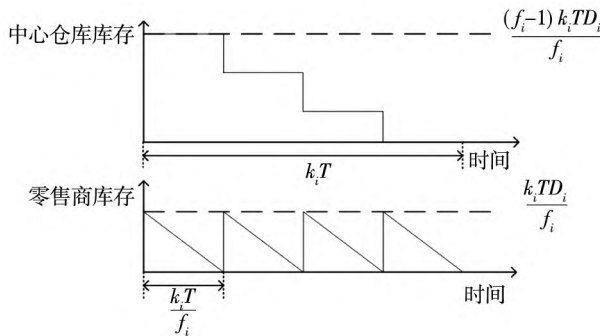


图2 库存水平变化示意图

Fig. 2 Cycle process of inventory level

订货成本  $C_o$  包括主要订货费用和次要订货费用,在每个基本周期会产生一次主要订货费用,而每个商品  $i$  则每隔  $k_i T$  的周期产生一次次要订货费用,因此订货成本为  $S/T + \sum_{i \in N} s_i^w / k_i T$ .

中心仓库的库存成本  $C_h^w$  受商品在中心仓库的平均库存水平影响.根据图2可知,每个商品  $i$  在一个库存周期内的平均库存水平为  $(f_i - 1) k_i T D_i / 2 f_i$ , 因此中心仓库的库存成本为  $\sum_{i \in N} (f_i - 1) k_i T D_i h_i^w / 2 f_i$ .

$k_i T D_i h_i^w / 2 f_i$ .

配送成本  $C_d$  受中心仓库向下游配送的频率影响,每次发生配送行为时产生一笔固定费用  $s_i^r$ ,每隔  $k_i T / f_i$  的时间发生一次配送,因此配送成本为  $\sum_{i \in N} f_i s_i^r / k_i T$ .

零售商的库存成本  $C_h^r$  受商品在零售商处的平均库存水平影响,每个商品  $i$  在一个库存周期内的平均库存水平为  $k_i T D_i / 2 f_i$ , 因此零售商处的库存成本为  $\sum_{i \in N} k_i T D_i h_i^r / 2 f_i$ .

综上,总成本函数如式(1)所示

$$TC(T, K, F) = C_o + C_h^w + C_d + C_h^r$$

$$= \left( \frac{S}{T} + \sum_{i \in N} \frac{s_i^w}{k_i T} \right) + \left( \sum_{i \in N} \frac{(f_i - 1) k_i T D_i h_i^w}{2 f_i} \right) + \left( \sum_{i \in N} \frac{f_i s_i^r}{k_i T} \right) + \left( \sum_{i \in N} \frac{k_i T D_i h_i^r}{2 f_i} \right) \quad (1)$$

基于以上分析,可得到 JRD 的数学模型如下

$$\min TC(T, K, F) \quad (2)$$

$$\text{s. t. } \begin{cases} T \in (0, +\infty) \\ K \in \mathbb{Z}_+^n \\ F \in \mathbb{Z}_+^n \end{cases} \quad (3)$$

如果令  $TC_i$  表示仅与商品  $i$  相关的独立成本,

即

$$TC_i(T, k_i, f_i) = \frac{s_i^w}{k_i T} + \frac{(f_i - 1) k_i T D_i h_i^w}{2 f_i} + \frac{f_i s_i^r}{k_i T} + \frac{k_i T D_i h_i^r}{2 f_i} \quad (4)$$

则总成本函数可以表示为

$$TC(T, K, F) = \frac{S}{T} + \sum_{i \in N} TC_i$$

从而, JRD 问题的混合整数规划模型也可以写作

$$(P) \min_{T, K, F} \left\{ \frac{S}{T} + \sum_{i \in N} TC_i(T, k_i, f_i) \mid \begin{cases} T > 0 \\ K \in \mathbb{Z}_+^n \\ F \in \mathbb{Z}_+^n \end{cases} \right\} \quad (5)$$

### 3 模型性质分析

下面对 JRD 模型的数学性质进行分析,为了便于表达,用  $\lceil \cdot \rceil$  表示向上取整函数,并规定  $\bar{A}$  和  $\underline{A}$  分别表示变量  $A$  的上界和下界.

**性质 1** (分解性质) 当基本补货周期  $T$  确定时, 可以通过  $n$  个子问题确定最优的  $(\mathbf{K}, \mathbf{F})$ .

**证明** 当基本补货周期  $T$  确定时, 商品之间的耦合关系将会解除, 此时每一个  $(k_i, f_i)$  都只与商品  $i$  相关, 问题 (P) 能够被分解为关于商品  $\{1, \dots, n\}$  的  $n$  个子问题. 问题 (P) 可等价转写为式 (6), 此时只需要分别求解每个子问题就可以确定最优的  $(\mathbf{K}, \mathbf{F})$

$$\min_{T>0} \left\{ \frac{S}{T} + \sum_{i \in N} \min_{k_i, f_i} \left\{ TC_i(T, k_i, f_i) \mid \begin{matrix} k_i \in \mathbb{Z}_+ \\ f_i \in \mathbb{Z}_+ \end{matrix} \right\} \right\} \quad (6)$$

证毕.

**性质 2** 当  $(\mathbf{K}, \mathbf{F})$  给定时, 最优的  $T$  满足

$$T^* = \sqrt{\frac{2 \left( S + \sum_{i \in N} \frac{s_i^w + f_i s_i^r}{k_i} \right)}{\sum_{i \in N} \left( h_i^w + \frac{h_i^r - h_i^w}{f_i} \right) k_i D_i}} \quad (7)$$

**证明** 当  $(\mathbf{K}, \mathbf{F})$  给定时, 原问题 (P) 退化为一个单变量连续优化问题, 此时只需要检查其一阶和二阶最优条件即可. 其二阶偏导数为

$$\frac{\partial^2 TC}{\partial T^2} = \frac{2S}{T^3} + \sum_{i \in N} \frac{2(s_i^w + f_i s_i^r)}{k_i T^3} > 0$$

因此, 当  $(\mathbf{K}, \mathbf{F})$  给定时, 函数  $TC$  是严格凸函数, 只需要满足一阶最优条件, 即可得到最优的  $T^*$ .

证毕.

**性质 3** 决策变量  $\mathbf{K}$  和  $\mathbf{F}$  都是有界的, 分别存在界限  $[1, \bar{K}]$  和  $[1, \bar{F}]$ .

**性质 3** 主要基于两方面的考量: 1) 现实因素. 由于决策变量  $\mathbf{K}$  和  $\mathbf{F}$  分别影响补货频率和配送频率, 这在现实中不可能是没有限制的, 在过往的研究<sup>[10, 16]</sup>中其上界往往限制为 20; 2) 理论因素. 即使没有给  $\mathbf{K}$  和  $\mathbf{F}$  赋予一个初始上界, Wang 等<sup>[10]</sup>的研究也证明了可以在  $O(n)$  时间内找到最优的  $\mathbf{K}$  和  $\mathbf{F}$  的上界.

通过性质 3 和性质 1 可以得到决策变量  $T$  的上下界如下

$$\bar{T} = \sqrt{\frac{2 \left( S + \sum_{i \in N} \frac{s_i^w + f_i s_i^r}{k_i} \right)}{\sum_{i \in N} \left( h_i^w + \frac{h_i^r - h_i^w}{f_i} \right) k_i D_i}} \quad (8)$$

$$\underline{T} = \sqrt{\frac{2 \left( S + \sum_{i \in N} \frac{s_i^w + f_i s_i^r}{k_i} \right)}{\sum_{i \in N} \left( h_i^w + \frac{h_i^r - h_i^w}{f_i} \right) k_i D_i}} \quad (9)$$

**性质 4** 给定一个  $T$  的范围  $[\underline{T}, \bar{T}]$ , 问题 (P) 的目标函数是 Lipschitz 连续的.

**证明** 若定义函数  $g_i(T) := \min \{ TC_i(T, k_i, f_i) : k_i \in \mathbb{Z}_+, f_i \in \mathbb{Z}_+ \}$ , 则式 (6) 可以进一步简化为  $\min \{ S/T + \sum_{i \in N} g_i(T) \}$ , 此时式 (6) 被分为了两部分: 所有商品共同分摊的主要订货成本  $S/T$ ; 商品自身的库存成本  $\sum_{i \in N} g_i(T)$ . 显然, 若  $l_0$  是  $S/T$  的 Lipschitz 常数且  $l_i$  是  $g_i(\cdot)$  的 Lipschitz 常数, 那么就找到了满足条件的 Lipschitz 常数  $l = l_0 + \sum_{i \in N} l_i$ . 既然  $S/T$  是可导的,  $l_0 = |S/T^2|$  是显然的, 只需要确定  $l_i$ . 容易证明, 函数  $TC_i(T, k_i, f_i)$  在  $T$  的定义域上是严格凸的, 其图像为一个 U 型曲线, 而根据定义,  $g_i(T)$  是  $TC_i(T, k_i, f_i)$  在不同的  $(k_i, f_i)$  取值时的最小值, 那么  $g_i(T)$  是一个由不同的  $TC_i(T, k_i, f_i)$  组成的分段函数, 尽管函数  $g_i(T)$  是不可微的, 但是在不同  $TC_i(T, k_i, f_i)$  图像的交点处, 其存在左导数和右导数的最大值, 这意味着其 Lipschitz 常数是存在的. 又由于性质 3, 决策变量  $\mathbf{K}$  和  $\mathbf{F}$  都是有界的, 那么可以用一个上界来估计 Lipschitz 常数, 即

$$l_i = \sup \left\{ \frac{dTC_i}{dT} \mid \begin{matrix} T \in [\underline{T}, \bar{T}] \\ k_i \in \{1, \dots, \bar{k}_i\} \\ f_i \in \{1, \dots, \bar{f}_i\} \end{matrix} \right\} \quad (10)$$

综上, 找到了满足条件的 Lipschitz 常数  $l = l_0 + \sum_{i \in N} l_i$ .

证毕.

**性质 5** 给定  $\mathbf{F}$  和  $T$ , 对应的最优的  $\mathbf{K}$  满足

$$k_i^* = \left\lceil -\frac{1}{2} + \frac{1}{2} \sqrt{1 + \frac{8f_i(s_i^w + f_i s_i^r)}{T^2 D_i [(f_i - 1) h_i^w + h_i^r]}} \right\rceil \quad (11)$$

**性质 6** 给定  $\mathbf{K}$  和  $T$ , 对应的最优的  $\mathbf{F}$  满足

$$f_i^* = \left\lceil -\frac{1}{2} + \frac{1}{2} \sqrt{1 + \frac{2(k_i T)^2 D_i (h_i^r - h_i^w)}{s_i^r}} \right\rceil \quad (12)$$

**证明** 性质5和性质6的证明较为简单,思路相同,这里仅简要介绍性质5的证明.可以发现最优的  $K$  的必要条件为  $TC_i(k_i) \leq TC_i(k_i - 1)$  且  $TC_i(k_i) \leq TC_i(k_i + 1)$ . 将其展开可以得到

$$k_i^2 + k_i - \frac{2f_i(s_i^w + f_i s_i^r)}{T^2 D_i[(f_i - 1)h_i^w + h_i^r]} \geq 0 \quad (13)$$

$$k_i^2 - k_i - \frac{2f_i(s_i^w + f_i s_i^r)}{T^2 D_i[(f_i - 1)h_i^w + h_i^r]} \leq 0 \quad (14)$$

求解两个不等式即可得到最优的  $k_i^*$ . 证毕.

## 4 求解算法设计

### 4.1 $\varepsilon$ -最优算法设计

本文提出的  $\varepsilon$ -最优算法思路如下:通过定界启发式算法缩小决策变量的搜索范围;将缩小后的决策变量界限输入到 Lipschitz 优化算法,通过动态计算 Lipschitz 常数找到最优解.具体定界启发式算法和 Lipschitz 优化算法的内容分析如下.

#### 4.1.1 定界启发式算法

对于 JRD 问题来说,问题涉及的决策变量有  $K$ 、 $F$ 、 $T$  三部分,搜索空间非常庞大,并且由于  $K$ 、 $F$  两个变量之间相互影响,其上界是难以确定的,常规的做法是直接设定其上界,  $K$ 、 $F$  的上界常被设定为 20,远大于 Cha 等<sup>[3]</sup>得到的最优解的 5 倍,由此来确保搜索空间包含了最优解,在既有研究中这种设定都得到了很好的应用效果,但是当问题的规模扩大时,寻优空间呈指数级上升,例如商品个数是 50 个时,搜索空间便达到了  $20^{100}$  量级,此时算法往往难以取得比较好的效果,这里借鉴文献[10]的思想,提出一种定界启发式算法(bound heuristic, BH),用于缩小搜索空间.

注意到,式(11)和式(12)给出了最优的  $K$ 、 $F$  的必要条件,进而可以得到最优的  $K$ 、 $F$  的界限公式,可以发现,  $K$ 、 $F$ 、 $T$  的界限相互之间只要确定了两者,便可以确定第三者,这种特殊的结构使得可以在给出  $K$ 、 $F$  的初始界限 20 后,可以通过循环迭代的方式,来得到一个更好的界限.具体而言,算法的主要思想是通过一个起始  $K$ 、 $F$  的界限,得到一个较好的  $T$  的界限,然后根据  $F$ 、 $T$  的界限得到一个更好的  $K$  的界限,利用新的  $K$  的界限和  $T$  的界限得到一个更好的  $T$  的界限,重复执

行这个过程,通过循环迭代,直到  $K$ 、 $F$  的界限不再变化.具体算法流程如图 3 所示.

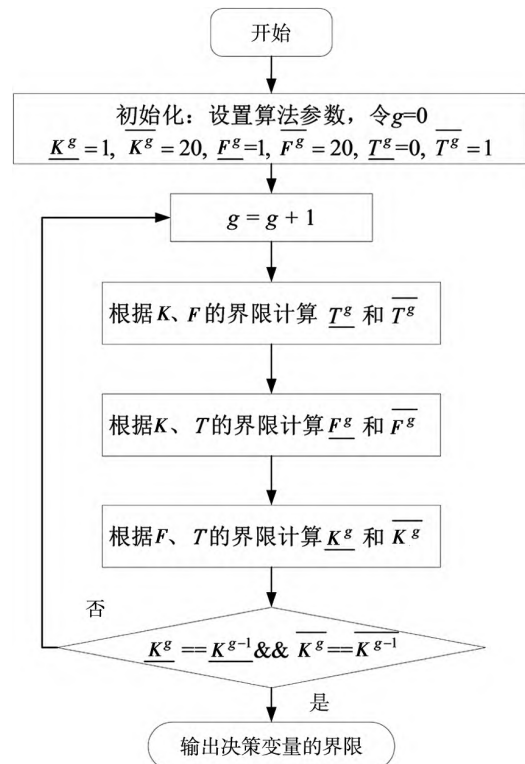


图3 定界启发式算法流程图

Fig. 3 The flow chart of the bound heuristic

#### 4.1.2 Lipschitz 优化算法

Lipschitz 优化算法是一类  $\varepsilon$ -最优算法, Lipschitz 优化算法在网络规划、非凸规划、多目标规划等领域应用广泛<sup>[30-33]</sup>, Lipschitz 优化相关的基础理论和一些代表性算法可以参考 Hansen 等<sup>[34, 35]</sup>的综述性文章. 本文选择 Evtushenko 算法来求解 JRD 问题,这主要是由于 JRD 问题的 Lipschitz 常数可以动态地计算,非常适合采用 Evtushenko 算法,并且 Evtushenko 算法的实现简单且效率较高.

Evtushenko 算法是一种序列式算法,算法对于自变量评价点的序列  $x_g$  是一个大小有序的序列,即下一个评价点的值比上一个评价点大,算法的过程是对自变量  $x$  的定义域  $[a, b]$  从左往右不断地寻找评价点,其基本思想是利用当前已知的最优的目标值来动态地决定下一次迭代的步长, Evtushenko 算法每次迭代的步长为  $\delta = (2\varepsilon + f_{opt} - f(x_g))/l$ , 其中  $f_{opt}$  是当前找到的最优值,  $x_g$  是第  $g$  次迭代的选取点,这样就使得下一个迭代

点的目标值下界不差于当前的最优值  $f_{opt}$ , 同时与最优解的偏差也不会超过  $\varepsilon$ . 由于 Evtushenko 算法是通过问题当前已知的最优的目标值与当前评价点的差距, 从而动态地决定下一次迭代的步长. 因此, 通过得到 JRD 问题的一个较好的上界, 可以大幅加快 Evtushenko 算法初期的迭代过程. 本文提出一种高效的启发式算法来获得 JRD 问题的一个上界, 其思路是利用式(7)、式(11)和式(12)实现三种决策变量之间的循环迭代, 算法伪代码如算法 1 所示. 整个 Evtushenko 算法的框架见算法 2.

算法 1: Simple Heuristic 算法

```

 $g \leftarrow 0$  //设置迭代数
 $(k_1^g, \dots, k_n^g) \leftarrow 1$  // 设置  $K$  的初值
 $(f_1^g, \dots, f_n^g) \leftarrow 1$  // 设置  $F$  的初值
 $T^g = 0$  //设置  $T$  的初值
 $g \leftarrow g + 1$ 
通过  $(k_1^{g-1}, \dots, k_n^{g-1})$  和  $(f_1^{g-1}, \dots, f_n^{g-1})$  计算  $T^g$ 
While  $T^g \neq T^{g-1}$ 
通过  $(f_1^{g-1}, \dots, f_n^{g-1})$  和  $T^g$  计算  $(k_1^g, \dots, k_n^g)$ 
通过  $(k_1^g, \dots, k_n^g)$  和  $T^g$  计算  $(f_1^g, \dots, f_n^g)$ 
 $g \leftarrow g + 1$ 
通过  $(k_1^{g-1}, \dots, k_n^{g-1})$  和  $(f_1^{g-1}, \dots, f_n^{g-1})$  计算  $T^g$ 
EndWhile

```

算法 2: Evtushenko 算法 (EV)

```

通过 BH 算法确定  $[\underline{T}, \bar{T}]$ 
 $g \leftarrow 1$ 
 $x_1 \leftarrow \underline{T} + \varepsilon/l$ 
 $x_{opt} \leftarrow x_h$  //通过算法 1 得到一个较好的解
 $TC_{opt} \leftarrow TC(x_{opt})$ 
While  $x_g < \bar{T}$  do
 $x_{g+1} \leftarrow x_g + (TC(x_g) - TC_{opt} + 2\varepsilon)/l$ 
If  $TC(x_{g+1}) < TC_{opt}$  then
 $TC_{opt} \leftarrow TC(x_{g+1})$ 
 $x_{opt} \leftarrow x_{g+1}$ 
EndIf
 $g \leftarrow g + 1$ 
EndWhile

```

算法 2 的收敛速度为  $O(1/\varepsilon)$ , 也就是说给定一个预设的误差上限  $\varepsilon$ , 需要  $O(1/\varepsilon)$  阶的迭代步数. 而关于算法复杂度, 由于 BH 算法的复杂度取决于问题参数, 因此复杂度较难分析, 但如果从实际应用出发, 只要加上两点改动, 该算法就能

变成时间复杂度为  $O(n)$  的算法: 1) 给定一个  $T$  的初始界限; 2) 采用预先设定的固定步长  $\delta$ . 从现实角度出发,  $T$  的界限往往是有限的, 例如有些文献中会将其固定为  $(0, 1]$ , 这是由于在现实中补货间隔显然不可能过长或过短. 固定步长  $\delta$  则是由于现实中补货间隔不可能无限制的短, 现有研究也有将最小补货间隔设为约束的问题, 假如时间单位为年, 那么可以设置最小步长为 0.000 1 (约 0.87 h), 这已经完全能满足实际需求. 在这两点改动下, 算法的时间复杂度只与商品种类数相关, 显然为  $O(n)$ .

#### 4.2 对偶下界及其求解算法

本文通过构造对偶问题提出一个原问题 (P) 的一个下界. 注意到, 如果将所有的商品进行独立补货而不是联合补货, 并将主要订货成本  $S$  分配给每一个商品, 即用  $\alpha_i S + s_i^w$  来替代原来的  $s_i^w$ , 其中  $\alpha_i \geq 0 \forall i \in N$  并且  $\sum_{i \in N} \alpha_i = 1$ , 令  $\alpha := (\alpha_1, \alpha_2, \dots, \alpha_n)$ , 则可以得到如下的独立补货问题 (P- $\alpha$ )

$$(P-\alpha) \min TC_\alpha = \sum_{i \in N} \frac{\alpha_i S + s_i^w}{T_i} + \sum_{i \in N} \frac{(f_i - 1) T_i D_i h_i^w}{2 f_i} + \sum_{i \in N} \frac{f_i s_i^r}{T_i} + \sum_{i \in N} \frac{T_i D_i h_i^r}{2 f_i} \quad (15)$$

$$\text{s. t. } \begin{cases} T_i \in (0, +\infty) \forall i \in N \\ F \in \mathbb{Z}_+^n \end{cases} \quad (16)$$

可以证明对于任意的  $\{\alpha: \alpha_i \geq 0 \forall i \in N, \sum_{i \in N} \alpha_i = 1\}$ , 总有  $\min TC_\alpha \leq \min TC$  成立: 不难发现, 对于问题 (P) 的一个可行解  $(T, K, F)$ , 令  $T_i = k_i T \forall i \in N$ ,  $F$  保持不变, 显然就得到了问题 (P) 的一个可行解  $(T_i, F)$ , 则有

$$TC - TC_\alpha = \frac{S}{T} - \sum_{i \in N} \frac{\alpha_i S}{T_i} = \frac{S}{T} - \sum_{i \in N} \frac{\alpha_i S}{k_i T} \geq \frac{S}{T} - \sum_{i \in N} \frac{\alpha_i S}{T} = 0 \quad (17)$$

式(17)的第一步等式变换源于  $TC$  与  $TC_\alpha$  的定义, 第二步等式变换源于  $T_i = k_i T$ , 第三步的不等式变换源于  $k_i \geq 1$ , 最后一步等式变换源于  $\sum_{i \in N} \alpha_i = 1$ .

如果定义  $D(\alpha) := \min TC_\alpha$ , 那么集合



$\{D(\alpha): \alpha_i \geq 0, \sum_{i \in N} \alpha_i = 1\}$  就定义了一类  $TC$  的下界,问题是找到其中最好的那个下界,因此可以得到原问题 (P) 的一个对偶问题 (P-D)

$$(P-D) \max_{\alpha} \{D(\alpha): \sum_{i \in N} \alpha_i = 1, \alpha_i \geq 0 \forall i \in N\} \quad (18)$$

对于已经确定  $\alpha$  取值的问题 (P- $\alpha$ ),其求解是非常简单的,相当于求解  $n$  个独立补货的子问题,商品之间不存在耦合关系,因此对偶问题的求解难点主要在于寻找最优的  $\alpha$  的值,不难发现将主要订货成本  $S$  分配给越小的  $T_i$ ,目标函数也能增加的更多,因此可以不断地进行这种分配直到不能分配为止,基于这种思想,这里提出一种启发式方法求解问题 (P-D),其步骤主要如下:

- 1) 设置所有的  $\alpha_i$  为 0;
- 2) 求解问题 (P- $\alpha$ ) 得到  $T_i$ ;
- 3) 找到所有  $T_i$  中最小者,对于相应的  $\alpha_i$  增加一定的量(如 0.000 1);
- 4) 重复 2)~3) 直到  $\sum_{i \in N} \alpha_i = 1$ .

## 5 数值实验与结果分析

为了验证算法的正确性和测试算法性能,本节设计了几组实验:1) 基于小规模算例的算法正

确性测试;2) 500 个随机算例的大规模算法性能测试;3) 对相关算法参数的影响进行分析;4) 对协同配送效益进行分析.所有算法采用 Matlab 2019b 进行编码,算例都在一台配置为 Intel Core i5-8600K 3.6GHz CPU 16GB RAM 且操作系统为 Windows10 的个人电脑中完成.

### 5.1 算法正确性检验

本文采用了文献[10]中的一个小规模算例来验证 Lipschitz 优化算法和对偶下界算法的正确性,数据如表 3 所示. EV 算法的容差  $\varepsilon$  设置为 0.1,运算结果如表 4 所示.

表 3 文献[12]的一个小规模算例 ( $S = 200$ )

Table 3 A small-scale instance from [12] ( $S = 200$ )

商品	1	2	3	4	5	6
$D_i$	10 000	5 000	3 000	1 000	600	200
$s_i^w$	45	46	47	44	45	47
$s_i^r$	5	5	5	5	5	5
$h_i^w$	1	1	1	1	1	1
$h_i^r$	1.5	1.5	1.5	1.5	1.5	1.5

从表 4 可以发现, EV 算法的结果与文献[10]提供的最优解保持一致,并且 EV 算法与 LB 算法的求解效率都很高.值得注意的是,下界算法得到了比文献[12]更好的结果,并且该结果与最优解很接近,  $Gap$  值仅为 0.22%,基本可以视为最优解.

表 4 EV 算法与对偶下界在小规模算例上的表现

Table 4 The performance of EV and dual lower bound on the small-scale instance

算法	目标值	运算时间	求解结果
EV	4 828.89	<0.01	$K = [1; 1; 1; 2; 2; 4]$ , $F = [4; 3; 2; 3; 2; 2]$ , $T = 0.188 1$
LB	4 818.26 ( $Gap = 0.22\%$ )	<0.01	—
文献[10]最优解	4 828.89	—	$K = [1; 1; 1; 2; 2; 4]$ , $F = [4; 3; 2; 3; 2; 2]$ , $T = 0.188 1$
文献[10]的下界	4 754.46 ( $Gap = 1.54\%$ )	—	—

### 5.2 大规模算例对比实验

本文设计了  $n = 10, 30, 50, 70, 100$  五组大规模实验,主要订货成本  $S$  设定为次要订货成本之和的 0.1 倍~0.5 倍,每组配置随机生成 20 个算例,这样总共生成了 500 个随机算例,其他实验数据生成规则参考文献[10]如表 5 所示.

表 5 算例规则

Table 5 The range of parameters

$s_i^w$	$D_i$	$s_i^r$	$h_i^w$	$h_i^r$
[40, 100]	[500, 5 000]	[0.1 $s_i^w$ , 0.3 $s_i^w$ ]	[0.5, 3]	[1.2 $h_i^w$ , 2.0 $h_i^w$ ]

为验证 EV 算法的性能,本文选择自适应差分进化算法 (adaptive differential evolution, ADE)、自适应混合差分进化算法 (adaptive hybrid

differential evolution, AHDE)、遗传算法 (genetic algorithm, GA) 进行对比, 原因如下: 既有的许多研究 (见表 1) 已经证实差分进化算法及其变种的各种算法在解决 JRD 问题时, 具有超过其他元启发式算法的准确性和稳定性, 尤其是 AHDE 在小规模算例中取得了非常好的效果, 是目前 JRD 问题上表现最好的元启发式算法之一; GA 则作为一种经典的群体智能算法而被广泛使用, 在文献

[3] 中也是被用于求解 JRD 取得了非常好的效果. 所有元启发式算法的最大迭代次数设置为 500, 种群大小设置为商品数的 10 倍, 其他具体的算法设计和参数可以参考文献 [3, 10, 18], 这里不再赘述. 统计 4 种求解算法以及对偶下界的计算结果, 如表 6 所示, 此外本文还统计了算法运行时间和算法准确度随算例参数变化的情况, 如图 4 和图 5 所示.

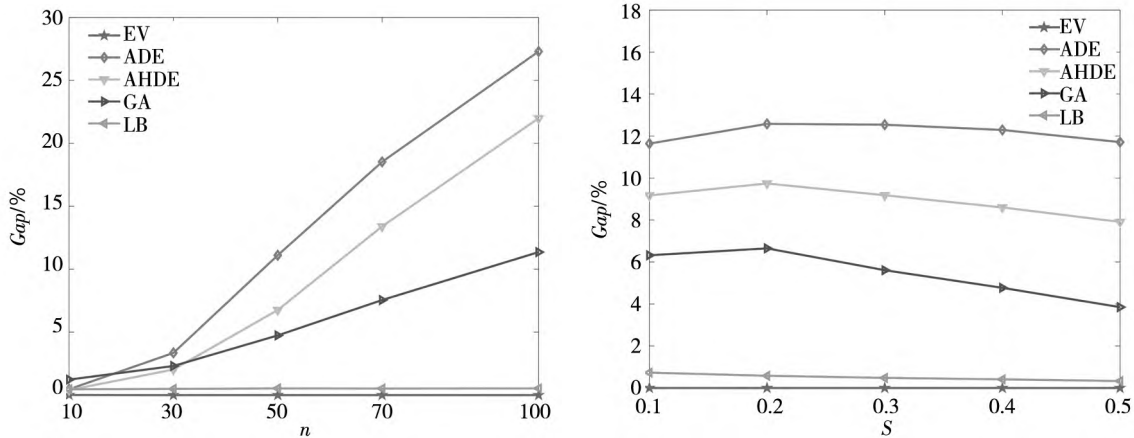


图 4 算例参数对算法准确性的影响

Fig. 4 Influence of instance parameters on algorithm accuracy

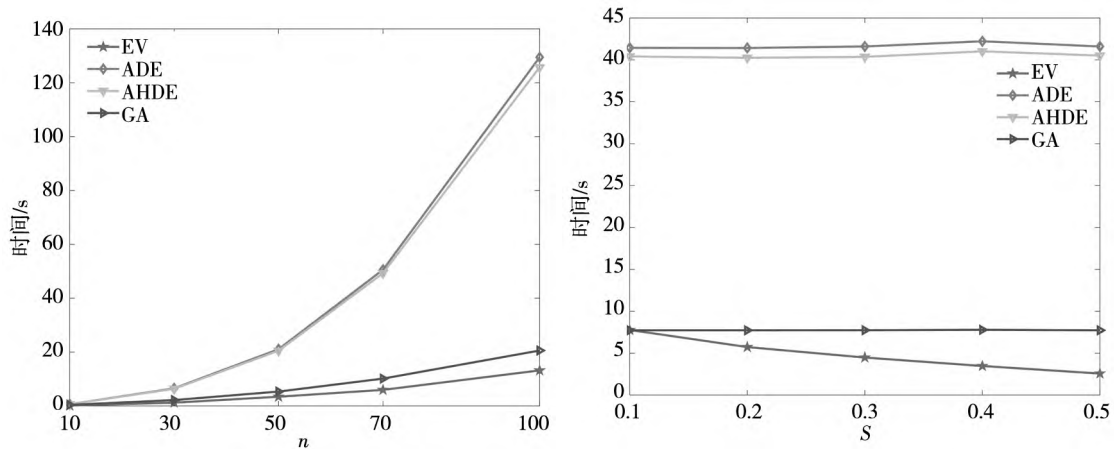


图 5 算例参数对算法运行时间的影响

Fig. 5 Influence of instance parameters on running time

从表 6 和图 4、图 5, 可以获得以下结论:

1) EV 算法展现出了非常好的准确度, 尤其是问题规模较大时. 由于本文设置的容差上限为  $\varepsilon = 0.1$ , 即算法所得目标值与最优值的偏差不会超过 0.1, 基本上可以认为是最优解, 在数值实验中 EV 算法得到解始终是几种算法中最优的. 与 EV 算法相比, 其他的元启发式算法尽管在小规模算例中取得了比较好的效果, 但在大规模算例

上都表现不佳, 例如, AHDE 算法在  $n = 10$  时, 经常能求到最优解, 这与既有研究的结论一致, 但当问题规模变大时, AHDE 算法的准确度明显下降, 在  $n = 100$  时, 目标值与最优值的偏差最大达到了 25.46%. 相比之下 GA 的鲁棒性则更好, 最大偏差是 14.13%. ADE 算法的最大偏差则达到了 31.37%. 这部分结论对既有算法在大规模算例上的仿真情况进行了有效的补充.

2) EV 算法的运行时间较短. 与几种元启发式算法相比, EV 算法随问题规模扩大而增长的幅度较小, 即使是  $n = 100$  的算例, 平均计算时间也不超过 23 s, 而 ADE 与 AHDE 则需要 120 s 以上, 相比之下 GA 随问题规模增长的幅度则要小得多,  $n = 100$  的算例只需要 20 s 左右.

3) 对偶下界与最优值偏差很小. 在 500 个算例中, 对偶  $Gap$  最大为 1.13%, 平均为 0.51%, 这一结果要优于文献 [10] 对 JRD 下界算法的研究结果, 并且本文的下界算法较为简单, 便于推广到一些 JRD 的扩展问题.

此外, 还可以发现以下的规律: 1) EV 算法的性能对于  $S$  的取值比较敏感. 当  $S$  的取值较大时, EV 算法的运行时间较短, 并且这一点在大规模算例上较为明显, 对于  $n = 100$  的算例,  $S$  取值 0.1 与 0.5 的情形, 平均运行时间能达到 3.4 倍的差距, 造成这一点主要是由于  $S$  较大时 Lipschitz 常数会偏小, 更有利于计算; 2) 对偶下界对于参数  $n$  和  $S$  的取值有些许依赖.  $n$  越大, LB 算法的  $Gap$  越大,  $S$  越大, LB 算法的  $Gap$  越小, 但这种影响非常小,  $n$  从 10 增加到 100, 平均  $Gap$  从 0.46% 增加到 0.53%,  $S$  从 0.1 增加到 0.5, 平均  $Gap$  从 0.73% 减少到 0.33%.

表 6  $Gap$  值、运算时间、最优次数的统计结果  
Table 6 Results of  $Gap$ , running time, and optimality number

$n$	$S$	$Gap$ 值/%										平均时间/s				最优次数			
		EV		ADE		AHDE		GA		LB		EV	ADE	AHDE	GA	EV	ADE	AHDE	GA
		平均	最大	平均	最大	平均	最大	平均	最大	平均	最大								
10	0.1	0.01	0.01	2.24	3.85	1.92	3.82	2.90	4.24	0.55	1.13	0.21	0.64	0.62	0.42	20	1	3	0
	0.2	0.01	0.01	0.07	1.11	0.07	1.11	2.05	6.73	0.55	0.98	0.21	0.64	0.62	0.41	20	15	15	1
	0.3	0.01	0.01	0.00	0.05	0.00	0.00	0.75	5.23	0.47	0.95	0.17	0.64	0.62	0.42	20	19	20	5
	0.4	0.01	0.01	0.00	0.07	0.00	0.04	0.29	1.24	0.44	0.74	0.15	0.64	0.62	0.42	20	18	18	7
	0.5	0.01	0.01	0.01	0.10	0.01	0.08	0.17	1.13	0.28	0.79	0.12	0.64	0.62	0.42	20	17	18	9
30	0.1	0.00	0.00	5.55	6.69	4.41	5.83	4.46	5.57	0.71	1.02	1.76	6.52	6.39	2.19	20	0	0	0
	0.2	0.00	0.00	5.58	6.71	3.92	6.42	4.04	6.81	0.55	0.93	1.42	6.51	6.36	2.19	20	0	0	0
	0.3	0.00	0.00	3.14	5.43	1.35	2.60	1.78	4.96	0.46	0.71	1.19	6.53	6.37	2.19	20	0	0	0
	0.4	0.00	0.00	1.58	2.75	0.29	0.87	0.69	2.48	0.40	0.64	1.00	6.56	6.39	2.20	20	0	0	0
	0.5	0.00	0.00	0.87	1.31	0.19	0.61	0.59	1.36	0.35	0.58	0.77	6.57	6.41	2.20	20	0	0	0
50	0.1	0.00	0.00	10.91	11.84	8.09	8.77	6.18	7.48	0.79	1.04	5.13	20.93	20.39	5.35	20	0	0	0
	0.2	0.00	0.00	11.80	13.24	8.31	9.07	6.41	7.54	0.63	0.82	4.57	21.04	20.44	5.35	20	0	0	0
	0.3	0.00	0.00	11.80	13.29	7.36	9.48	5.23	7.71	0.49	0.76	3.36	21.07	20.52	5.36	20	0	0	0
	0.4	0.00	0.00	11.32	14.08	5.90	8.17	3.84	5.93	0.41	0.61	2.32	21.12	20.61	5.36	20	0	0	0
	0.5	0.00	0.00	9.70	11.16	4.16	5.61	2.02	3.60	0.35	0.55	1.79	21.18	20.64	5.35	20	0	0	0
70	0.1	0.00	0.00	16.54	17.36	12.56	13.72	7.98	9.08	0.77	1.02	9.10	50.68	49.42	10.16	20	0	0	0
	0.2	0.00	0.00	18.62	20.16	14.14	15.35	8.73	9.72	0.61	0.76	6.69	50.49	49.23	10.15	20	0	0	0
	0.3	0.00	0.00	19.41	20.97	14.39	15.47	8.28	10.83	0.48	0.73	6.02	50.46	49.15	10.16	20	0	0	0
	0.4	0.00	0.00	19.45	21.29	13.56	15.22	7.27	9.30	0.42	0.62	4.57	50.53	49.19	10.17	20	0	0	0
	0.5	0.00	0.00	18.62	20.16	12.33	13.99	5.48	7.64	0.34	0.53	3.53	50.72	49.46	10.16	20	0	0	0
100	0.1	0.00	0.00	22.95	23.99	18.85	19.85	10.05	11.98	0.83	1.06	22.65	128.45	125.21	20.52	20	0	0	0
	0.2	0.00	0.00	26.83	28.94	22.25	23.48	12.00	13.46	0.58	0.80	15.78	128.40	124.59	20.53	20	0	0	0
	0.3	0.00	0.00	28.33	30.47	22.80	24.37	11.98	14.13	0.51	0.66	11.64	129.28	125.16	20.58	20	0	0	0
	0.4	0.00	0.00	29.08	30.41	23.25	25.46	11.77	13.54	0.40	0.54	9.31	132.23	128.26	20.80	20	0	0	0
	0.5	0.00	0.00	29.32	31.37	22.87	24.87	11.01	13.31	0.33	0.41	6.62	128.89	125.40	20.54	20	0	0	0

注: EV 算法  $Gap$  通过  $\varepsilon/\text{EV}$  计算, 其他算法  $Gap$  通过与 EV 算法结果比值得到.

5.3 算法参数的分析实验

算法自身的参数有两个: 1) 目标值的容差  $\varepsilon$ ;

2) 决策变量的上界. 本节选择了一个  $n$  为 100 的算例来测试两者对于算法性能的影响.

首先,测试容差  $\varepsilon$  的影响,统计了  $\varepsilon$  在  $[0.1, 1\ 000]$  范围内算法的表现,运行时间如图 6 所示,  $Gap$  值则如图 7 所示. 从图 6 来看,随着  $\varepsilon$  的增大,算法运行时间急剧下降,这一点符合对于算法收敛速度的分析:当其他参数都确定时,算法的收敛速度为  $O(1/\varepsilon)$ . 而从图 7 来看,  $\varepsilon$  对于算法准确度的影响比较有限. 当  $\varepsilon$  增大时,算法  $Gap$  值整体上会有上升的趋势. 但这种趋势并不绝对,呈现出一种波动的状态. 并且,即使是在  $\varepsilon = 1\ 000$  的情况下,算法依然获得了很高的准确度,  $Gap$  值基本可以忽略不计,远低于理论上的容差上限. 因此从运行时间和准确度两方面综合考虑,将  $\varepsilon$  设定为  $1 \sim 10$  就足以应对实际应用.

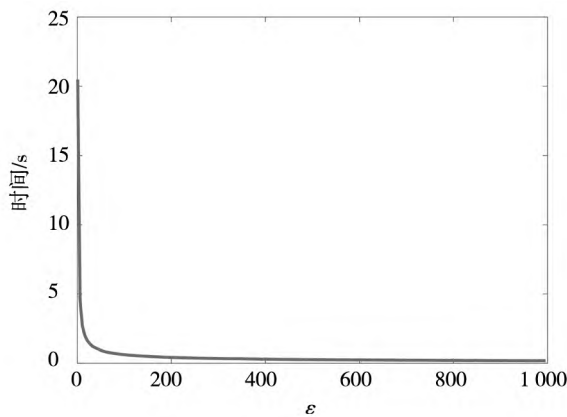


图 6  $\varepsilon$  对算法运行时间的影响

Fig. 6 Influence of  $\varepsilon$  on running time

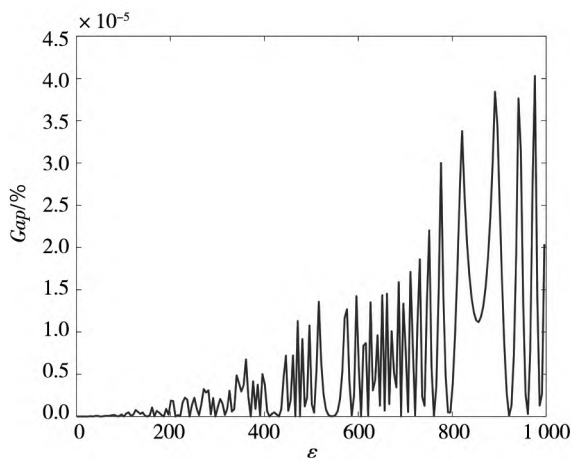


图 7  $\varepsilon$  对  $Gap$  值的影响

Fig. 7 Influence of  $\varepsilon$  on  $Gap$  value

其次,测试决策变量  $K$ 、 $F$  的上界的影响. 由于算法的基础假设是最优解存在于设定的决策变量界限内,如果上界设置的越大,那么找到最优解

的可能性也就越大,但计算时间也会增加. 因此,本文测试了决策变量上界在  $3 \sim 50$  的范围内的算法表现,运行时间如图 8 所示,  $Gap$  值则如图 9 所示. 从图 8 来看,随着决策变量上界的增加,运行时间的增长较为显著. 从图 9 来看,当上界设定为 5 以上时,算法便能够找到最优解. 进一步的增加上界的效果并不显著. 综合考虑,将上界设定为  $10 \sim 20$  便足以应对实际应用,这也与已有文献中的设定相符.

从本节的实验结果来看,当应用 Lipschitz 算法求解 JRD 问题时,  $\varepsilon$  设定为  $1 \sim 10$ , 决策变量上界设定为  $10 \sim 20$  就足以应对实际问题.

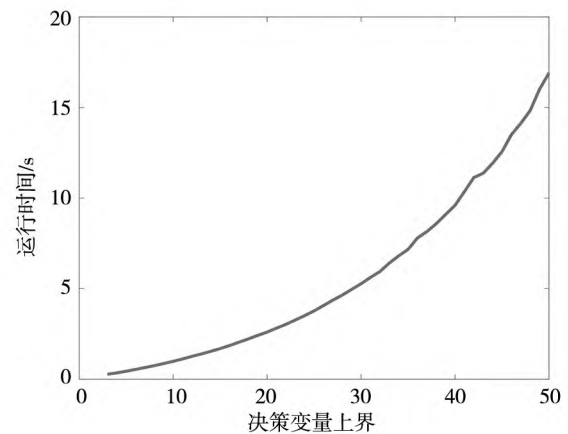


图 8 决策变量界限对算法运行时间的影响

Fig. 8 Influence of bounds of decision variables on running time

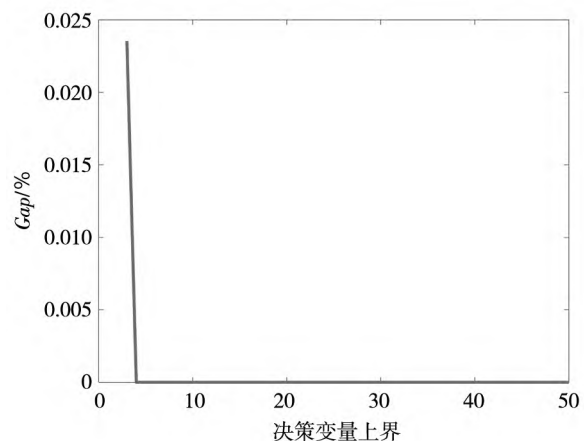


图 9 决策变量界限对  $Gap$  值的影响

Fig. 9 Influence of bounds of decision variables on  $Gap$  value

## 5.4 协同配送效益分析

JRD 与 JRP 最大的不同点就在于通过中心仓库来协调补货和配送决策,以此来达到更优的库存效率. 如果没有中心仓库来协同配送,那么就会

与基础 JRP 相同,在每一次补货的同时进行一次配送,此时的目标函数如式(19)所示.

$$TC_2 = \frac{S}{T} + \sum_{i \in N} \left( \frac{s_i^w + s_i^r}{k_i T} + \frac{k_i TD_i h_i^r}{2} \right) \quad (19)$$

通过 5.1 节给出的算例,可以分析协同配送策略的效益,所有计算都通过 EV 算法来完成,结果见表 7. 从表 7 可以发现:1)协同配送策略能带来一定的成本节约. 与无协同相比,协同配送策略会导致补货阶段的成本一定程度上升,但配送阶段的成本则会下降,整体上而言,协同配送策略会带来成本的节约; 2)协同配送策略下补货与配送周期有很大改变. 相比无协同的情况,协同配送

策略下,补货周期较长,而配送周期较短. 这主要是由于存在协同策略,企业可以更灵活的进行采购,并针对性地安排优化的配送作业计划.

综上所述,由于存在中心仓库的协同配送,补货与配送周期更为协调,配送策略会更加灵活,而无协同的 JRP 模型由于每次补货必须发生一次配送,导致很难根据每种商品的特点进行针对性的配送调度安排,造成配送阶段的周期要更长,成本也更高. 此外,由于中心仓库充当了缓冲器的作用,补货和配送决策实现了分离,降低了零售商与供应商进行协调的困难. 在这种模式下,即使不考虑整个供应链的集成优化,双方也可以独立进行调度安排.

表 7 协同与无协同的对比结果

Table 7 Comparison results of coordinated delivery and non-coordinated delivery

	补货阶段成本	配送阶段成本	总成本	$T$	$K$	$F$	协同效益
协同配送	3 475. 21	1 353. 68	4 828. 89	0. 188 1	1, 1, 1, 2, 2, 4	4, 3, 2, 3, 4, 2	7. 79 %
无协同	2 484. 44	2 752. 26	5 236. 70	0. 158 7	1, 1, 1, 2, 2, 4	—	—

为了进一步分析协同效益的敏感性,本节还进行了必要的敏感性分析. 首先对于每种参数进行单独的分析,测试  $S$ 、 $s_i^w$ 、 $s_i^r$ 、 $h_i^w$ 、 $h_i^r$  几种成本参数的单独变化对于协同效益的影响,实验结果如图 10 所示.

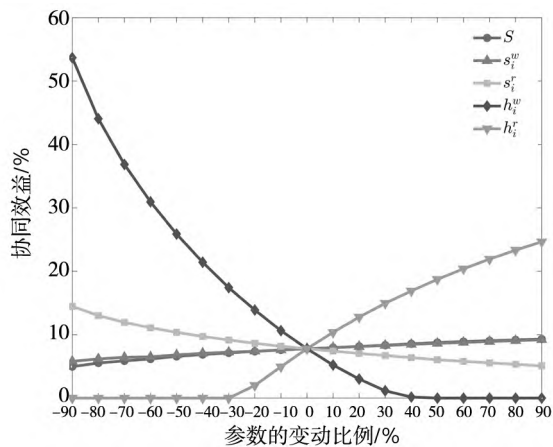


图 10 各种参数对协同效益的影响  
Fig. 10 Influence of parameters on coordination effect

从图 10 可以发现,  $h_i^w$  和  $h_i^r$  对协同效益的影响非常大,其他的几种成本参数的影响则比较小. 减小  $h_i^w$  或者是增大  $h_i^r$  都可以提升协同效益,相对来说,减小  $h_i^w$  的效果更加显著. 值得注意的一点是:当  $h_i^r$  减小幅度达到 30%,或者  $h_i^w$  增大幅度达

到 40%之后,协同效益几乎会变为 0. 为了进一步分析原因,将  $h_i^w$  和  $h_i^r$  对协同效益的联合影响进行了分析,结果如图 11 所示.

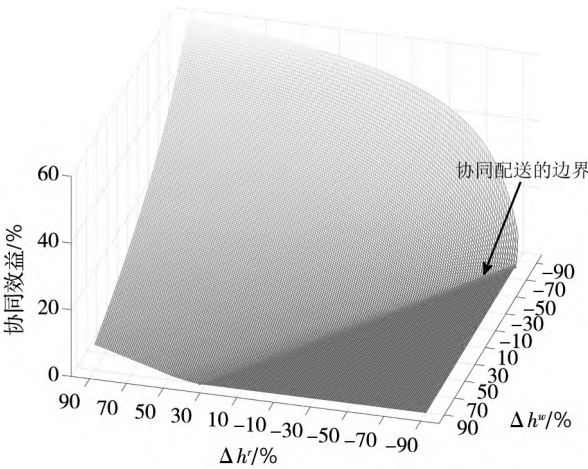


图 11  $h_i^w$  和  $h_i^r$  对协同效益的联合影响  
Fig. 11 Joint influence of  $h_i^w$  and  $h_i^r$  on coordination effect

从图 11 可以发现,该曲面右下角存在一条直线的边界,当  $(h_i^w, h_i^r)$  处在这条边界右方时,协同效益为 0,而处在左方时,协同效益则会迅速上升. 也就是说,存在一个协同配送策略的边界,在这个边界内,协同配送的 JRP 模型与无协同配送的 JRP 模型没有区别,此时进行协同配送策略将失去意义. 如果进一步分析图 11 的结果,可以发

现这条协同效益的边界上的点刚好满足  $h_i^w = h_i^r$ , 也就是说当中心仓库的单位库存成本等于零售商处的单位库存成本时, 协同效益为 0. 这说明当企业中心仓库与零售商处库存管理水平有明显差距时, 协同配送的效果会比较好, 反之协同配送策略起不到作用. 对于这个结论其实不难理解: 当中心仓库的单位库存成本与零售商处持平时, 零售商将没有足够的动力把产品储存在中心仓库, 而是会选择自行对商品进行储存, 那么此时中心仓库将会失去作用, 也就起不到协同配送的效果.

因此, 可得到一个有益的管理启示: 在企业采纳 JRD 模型, 必须要保证中心仓库拥有库存管理水平优势. 当中心仓库的库存管理水平比零售商处的库存管理水平有明显优势时, 协同配送的收益将会非常明显, 反之协同配送策略起不到节约成本的作用.

## 6 结束语

JRD 的研究一直缺少一个最优算法, 本文围绕其 Lipschitz 连续性首先提出了一种  $\varepsilon$ - 最优算法和对偶下界, 通过 500 个随机算例与既有的元启发式算法对比, 有效评估了算法的有效性, 在准确度和运算效率上, 算法都具有非常良好的表现. 本文还提出了 JRD 的一种多项式复杂度算

法: 只需要对基础补货周期设定一个界限, 例如  $(0, 1]$ , 具有最小步长限制的 Lipschitz 优化算法 (如 Passive 算法和 Evtushenko 算法) 的复杂度显然是  $O(n)$ . 此外, 进一步分析了协同配送策略的效益, 实验数据发现: 只有当中心库存成本有优势时, 协同配送才能够实现成本仓库的单位库存成本相较零售商处的单位节约, 并且随着这种优势的增加, 协同配送策略的效益会显著增长. 因此企业管理者在应用 JRD 模型时, 需要采取针对性改进措施, 提升中心仓库的库存管理水平.

未来可在以下方向进行深入研究: 1) JRD 的成本分配博弈问题. JRD 的成本分配博弈目前仍然是一个有待研究的问题, 这方面的研究属于合作博弈与物流管理交叉的领域, 已有一定的研究基础, 如文献[36]就提出了一种 B-T Shapley 值法对车辆路径问题进行成本分配; 2) 考虑其他更新颖的 JRD 扩展模型. 文献[22]提出了将技术投资和 JRD 进行集成优化的新问题, 这可以联系到供应商将技术投资与产品进行捆绑销售的背景<sup>[37]</sup>, 另外, 考虑配送过程中的逆向物流整合也是一个有趣的问题<sup>[38]</sup>; 3) 设计大规模复杂 JRD 问题的求解算法. 现实中存在一些成千上万物品种数的 JRD 问题, 此类问题目前传统方法都存在效率低下的问题, 而新颖的机器学习、深度学习方法有望在此领域取得突破<sup>[39]</sup>.

## 参 考 文 献:

- [1] Nilsson A, Silver E A. A simple improvement on silver's heuristic for the joint replenishment problem[J]. Journal of the Operational Research Society, 2008, 59(10): 1415 - 1421.
- [2] Khouja M, Goyal S. A review of the joint replenishment problem literature: 1989 - 2005[J]. European Journal of Operational Research, 2008, 186(1): 1 - 16.
- [3] Cha B C, Moon I K, Park J H. The joint replenishment and delivery scheduling of the one-warehouse, N-retailer system[J]. Transportation Research Part E: Logistics and Transportation Review, 2008, 44(5): 720 - 730.
- [4] 王 林, 顿彩霞, 张金隆. 基于混合差分进化算法的联合补货 - 配送优化模型[J]. 中国管理科学, 2012, (3): 86 - 93.  
Wang Lin, Dun Caixia, Zhang Jinlong. Integrated joint replenishment and distribution model using hybrid differential evolution algorithm [J]. Chinese Journal of Management Science, 2012, (3): 86 - 93. (in Chinese).
- [5] Porras E, Dekker R. A solution method for the joint replenishment problem with correction factor[J]. International Journal of Production Economics, 2008, 113(2): 834 - 851.

- [6] Hsu S-L. Optimal joint replenishment decisions for a central factory with multiple satellite factories[J]. *Expert Systems with Applications*, 2009, 36(2, Part 1): 2494–2502.
- [7] Zeng Y-R, Wang L. A hybrid decision support system for slow moving spare parts joint replenishment: A case study in a nuclear power plant[J]. *International Journal of Computer Applications in Technology*, 2010, 37(3–4): 287–296.
- [8] Wang L, He J, Zeng Y-R. A differential evolution algorithm for joint replenishment problem using direct grouping and its application[J]. *Expert Systems*, 2012, 29(5): 429–441.
- [9] Cohen-Hillel T, Yedidsion L. The periodic joint replenishment problem is strongly NP-hard[J]. *Mathematics of Operations Research*, 2018, 43(4): 1269–1289.
- [10] Wang L, Liu R, Liu S. Variable neighborhood search incorporating a new bounding procedure for joint replenishment and delivery problem[J]. *Journal of the Operational Research Society*, 2018, 69(2): 201–219.
- [11] Cha B C, Park J H. The joint replenishment and delivery scheduling involving multiple suppliers offering different quantity discounts[EB/OL]. //2009 International Conference on Computers & Industrial Engineering, Troyes, France, 2009: 52–56.
- [12] Liu R, Zeng Y-R, Qu H, et al. Optimizing the new coordinated replenishment and delivery model considering quantity discount and resource constraints[J]. *Computers & Industrial Engineering*, 2018, (116): 82–96.
- [13] Otero-Palencia C, Amaya-Mier R, Yie-Pinedo R. A stochastic joint replenishment problem considering transportation and warehouse constraints with gainsharing by shapley value allocation[J]. *International Journal of Production Research*, 2019, 57(10): 3036–3059.
- [14] Carvajal J, Castaño F, Sarache W, et al. Heuristic approaches for a two-echelon constrained joint replenishment and delivery problem[J]. *International Journal of Production Economics*, 2020, (220): 107420.
- [15] Moon I K, Cha B C, Lee C U. The joint replenishment and freight consolidation of a warehouse in a supply chain[J]. *International Journal of Production Economics*, 2011, 133(1): 344–350.
- [16] Cui L, Wang L, Deng J, et al. Intelligent algorithms for a new joint replenishment and synthetical delivery problem in a warehouse centralized supply chain[J]. *Knowledge-Based Systems*, 2015, (90): 185–198.
- [17] Ai X-Y, Zhang J-L, Wang L. Optimal joint replenishment policy for multiple non-instantaneous deteriorating items[J]. *International Journal of Production Research*, 2017, 55(16): 4625–4642.
- [18] Qu H, Wang L, Zeng Y-R. Modeling and optimization for the joint replenishment and delivery problem with heterogeneous items[J]. *Knowledge-Based Systems*, 2013, (54): 207–215.
- [19] Wang L, Peng L, Wang S, et al. Advanced backtracking search optimization algorithm for a new joint replenishment problem under trade credit with grouping constraint[J]. *Applied Soft Computing*, 2020, (86): 105953.
- [20] Ongkunaruk P, Wahab M I M, Chen Y. A genetic algorithm for a joint replenishment problem with resource and shipment constraints and defective items[J]. *International Journal of Production Economics*, 2016, (175): 142–152.
- [21] Chen Y, Yang L, Jiang Y, et al. Joint replenishment decision considering shortages, partial demand substitution, and defective items[J]. *Computers & Industrial Engineering*, 2019, (127): 420–435.
- [22] 崔利刚, 邓洁, 王林, 等. 基于改进联合采购及配送模型的 RFID 投资决策研究[J]. *中国管理科学*, 2018, 26(5): 86–97.
- Cui Ligang, Deng Jie, Wang Lin, et al. A study on the RFID investment decision model for the revised joint replenishment and delivery problem [J]. *Chinese Journal of Management Science*, 2018, 26(5): 86–97. (in Chinese)
- [23] Liu R, Liu S, Zeng Y-R, et al. Optimization model for the new coordinated replenishment and delivery problem with multi-warehouse[J]. *International Journal of Logistics Management*, 2017, 28(2): 290–310.
- [24] Yao M-J, Lin J-Y, Lin Y-L, et al. An integrated algorithm for solving multi-customer joint replenishment problem with distracting consideration[J]. *Transportation Research Part E: Logistics and Transportation Review*, 2020, (138): 101896.

- [25] Anily S, Haviv M. The cost allocation problem for the first order interaction joint replenishment model[J]. *Operations Research*, 2007, 55(2): 292–302.
- [26] He S, Sethuraman J, Wang X, et al. A non-cooperative approach to cost allocation in joint replenishment[J]. *Operations Research*, 2017, 65(6): 1562–1573.
- [27] 肖 旦, 周永务, 范丽繁, 等. 改良技术共享下改良品联合采购联盟的竞合博弈研究[J]. *中国管理科学*, 2019, 27(2): 129–137.
- Xiao Dan, Zhou Yongwu, Fan Lifan, et al. Co-opetition game analysis of joint purchasing coalitions of ameliorating items with ameliorating technology sharing[J]. *Chinese Journal of Management Science*, 2019, 27(2): 129–137. (in Chinese)
- [28] 张云丰, 王 勇, 龚本刚, 等. 考虑信用销售的改良品联合采购决策[J]. *中国管理科学*, 2018, 26(7): 84–96.
- Zhang Yunfeng, Wang Yong, Gong Bengang, et al. The decision-making of joint purchasing for amelioration items considering credit sales[J]. *Chinese Journal of Management Science*, 2018, 26(7): 84–96. (in Chinese)
- [29] 肖 旦, 姬晓楠, 罗 颜, 等. 基于服务竞争的远视零售商联合采购联盟的竞合博弈策略分析[J]. *中国管理科学*, 2024, 32(7): 270–280.
- Xiao Dan, Ji Xiaonan, Luo Yan, et al. Co-opetition game analysis of joint purchasing coalitions of farsighted retailers based on service competition[J]. *Chinese Journal of Management Science*, 2024, 32(7): 270–280. (in Chinese)
- [30] Fischione C. Fast-Lipschitz optimization with wireless sensor networks applications[J]. *IEEE Transactions on Automatic Control*, 2011, 56(10): 2319–2331.
- [31] Griewank A, Walther A, Fiege S, et al. On Lipschitz optimization based on gray-box piecewise linearization[J]. *Mathematical Programming*, 2016, 158(1): 383–415.
- [32] Gimbutas A, Žilinskas A. An algorithm of simplicial Lipschitz optimization with the bi-criteria selection of simplices for the bi-section[J]. *Journal of Global Optimization*, 2018, 71(1): 115–127.
- [33] 李红武, 谢 敏, 张 榕. 一类非光滑凸优化问题的邻近梯度算法[J]. *运筹学学报*, 2021, 25(1): 61–72.
- Li Hongwu, Xie Min, Zhang Rong. A proximal gradient method for nonsmooth convex optimization problems[J]. *Operations Research Transactions*, 2021, 25(1): 61–72. (in Chinese)
- [34] Hansen P, Jaumard B, Lu S-H. Global optimization of univariate Lipschitz functions: I. Survey and properties[J]. *Mathematical Programming*, 1992, 55(1–3): 251–272.
- [35] Hansen P, Jaumard B, Lu S-H. Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison[J]. *Mathematical Programming*, 1992, 55(1–3): 273–292.
- [36] 饶卫振, 朱庆华, 金 淳, 等. 协作车辆路径成本分摊问题的 B-T Shapley 方法[J]. *管理科学学报*, 2019, 22(1): 107–126.
- Rao Weizhen, Zhu Qinghua, Jin Cun, et al. A binary tree shapely method for cost sharing of the collaborative vehicle routing problem[J]. *Journal of Management Sciences in China*, 2019, 22(1): 107–126. (in Chinese)
- [37] 魏 航, 谈 丹, 李 佩. 具有技术转让的捆绑采购最优决策研究[J]. *管理科学学报*, 2016, 19(6): 1–19.
- Wei Hang, Tan Dan, Li Pei. Optimal bundled procurement with technology acquisition[J]. *Journal of Management Sciences in China*, 2016, 19(6): 1–19. (in Chinese)
- [38] 徐小峰, 姜明月, 邓忆瑞. 整合逆向物流协同配送动态路径优化问题研究[J]. *管理科学学报*, 2021, 24(10): 106–126.
- Xu Xiaofeng, Jiang Mingyue, Deng Yirui. Dynamic vehicle routing problem with simultaneous pickup and delivery in collaborative distribution under demand concurrent[J]. *Journal of Management Sciences in China*, 2021, 24(10): 106–126. (in Chinese)
- [39] 刘冠男, 曲金铭, 李小琳, 等. 基于深度强化学习的救护车动态重定位调度研究[J]. *管理科学学报*, 2020, 23



(2): 39–53.

Liu Guannan, Qu Jinming, Li Xiaolin, et al. Dynamic ambulance redeployment based on deep reinforcement learning[J]. Journal of Management Sciences in China, 2020, 23(2): 39–53. (in Chinese)

## An $\varepsilon$ -optimal algorithm and a dual lower bound for the joint replenishment and delivery problem

WANG Si-rui, WANG Lin<sup>\*</sup>, WU Bin-rong, ZHANG Jin-long

School of Management, Huazhong University of Science and Technology, Wuhan 430074, China

**Abstract:** This paper addresses the common joint replenishment and delivery (JRD) problem that occurs in a multi-item inventory system. An  $\varepsilon$ -optimal algorithm and a dual lower bound are proposed via deeply analyzing the mathematical properties of the model. The randomized numerical experiments show that the proposed  $\varepsilon$ -optimal algorithm outperforms the existing meta heuristics both in terms of both accuracy and efficiency, and the largest increase in accuracy can reach 31%. The dual lower bound is very tight, with the gap averaging below 0.84%. Moreover, the proposed  $\varepsilon$ -optimal algorithm is also very fast, and a 100-item instance can be solved in twenty-three seconds. In addition, the time complexity of the  $\varepsilon$ -optimal algorithm is analyzed. The  $\varepsilon$ -optimal algorithm can achieve  $O(n)$  polynomial-time complexity under real-world situations, which enriches the algorithmic design theory of the JRD research. Finally, the effect of coordinated delivery is analyzed. Experiments show that coordinated delivery can achieve cost reduction when the inventory management level of the central warehouse is higher than that of the retailers.

**Key words:** joint replenishment and delivery; Lipschitz optimization;  $\varepsilon$ -optimal; bound heuristic; dual lower bound